R Notebook

```
#data
data1<-read.table("/Users/phamkhoa/Documents/university/3/stats_model_1/w4/leukoplakia.txt", sep="\t", 
attach(data1)

data2<-read.table("/Users/phamkhoa/Documents/university/3/stats_model_1/w4/applejuiceCRA7152.txt", sep=
attach(data2)
```

```
#p1
#a
model.a<-glm(Leukoplakia~factor(Alcohol)+factor(Smoker), family = binomial("logit"), data1)
summary(model.a)
```

```
##
## Call:
## glm(formula = Leukoplakia ~ factor(Alcohol) + factor(Smoker),
##     family = binomial("logit"), data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9033  -1.1179   0.5974   0.9537   1.4694
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.08034    0.37488  -0.214  0.83031
## factor(Alcohol)2  0.63237    0.38896   1.626  0.10400
## factor(Alcohol)3 -0.06116    0.48852  -0.125  0.90038
## factor(Alcohol)4 -0.58411    0.72755  -0.803  0.42206
## factor(Smoker)1   1.08078    0.35051   3.083  0.00205 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 275.49  on 211  degrees of freedom
## Residual deviance: 255.93  on 207  degrees of freedom
## AIC: 265.93
##
## Number of Fisher Scoring iterations: 4
```

```
newdata<-data.frame(Alcohol = 3, Smoker = 1)
predict(model.a,newdata=newdata, type="response")
```

```
##         1
## 0.7189541
```

```r
# P(Y = 1) = 0.7189541
```

```r
#b
model.bH0<-glm(Leukoplakia~factor(Alcohol), family = binomial("logit"), data1)

model.bH1<-glm(Leukoplakia~factor(Alcohol)+factor(Smoker), family = binomial("logit"), data1)

anova(model.bH0, model.bH1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Leukoplakia ~ factor(Alcohol)
## Model 2: Leukoplakia ~ factor(Alcohol) + factor(Smoker)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       208     266.09
## 2       207     255.93  1   10.167  0.00143 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(model.bH0, model.bH1, test = "Chi")$Deviance[2]
```

```
## [1] 10.16722
```

```r
#10.16722
```

```r
#c
model.c<-glm(Leukoplakia~factor(Alcohol)+factor(Smoker), family = binomial("logit"), data1)
newdata<-data.frame(Alcohol = c(3,4), Smoker = c(1,1))
pred<-predict(model.c,newdata=newdata, type="response")
predict.data<-data.frame(newdata,pred)

OR<-(pred[2]/(1-pred[2]))/(pred[1]/(1-pred[1]))
OR
```

```
##         2
## 0.5927646
```

```r
#0.5927646
```

```r
#d
model.dH0<-glm(Leukoplakia~factor(Alcohol)+factor(Smoker), family = quasibinomial("logit"), data1)

model.dH1<-glm(Leukoplakia~factor(Alcohol)*factor(Smoker), family = quasibinomial("logit"), data1)

anova(model.dH0, model.dH1, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: Leukoplakia ~ factor(Alcohol) + factor(Smoker)
## Model 2: Leukoplakia ~ factor(Alcohol) * factor(Smoker)
##   Resid. Df Resid. Dev Df Deviance      F Pr(>F)
## 1       207     255.93
## 2       204     254.55  3   1.3718 0.4421 0.7232
```

```
anova(model.dH0, model.dH1, test = "F")$F[2]
```

```
## [1] 0.442091
```

```
# F = 0.442091
```

```
#e
model.eprobit<-glm(Leukoplakia~factor(Alcohol)+factor(Smoker), family = binomial("probit"), data1)
model.ecauchy<-glm(Leukoplakia~factor(Alcohol)+factor(Smoker), family = binomial("cauchit"), data1)
model.ecloglog<-glm(Leukoplakia~factor(Alcohol)+factor(Smoker), family = binomial("cloglog"), data1)

AIC(model.eprobit)
```

```
## [1] 265.9905
```

```
AIC(model.ecauchy)
```

```
## [1] 265.7605
```

```
AIC(model.ecloglog)
```

```
## [1] 266.3199
```

```
# Choose the cloglog link model
```

```
#p3
#a
# model g(u) = b0 +b1x1 + b2x2 +b3x3 +b4x4
#choose Ber distribution
#Response from R for each different link function in Binomial distribution
#identity: Error: no valid set of coefficients has been found: please supply starting values
#log: Error: no valid set of coefficients has been found: please supply starting values
#inverse: Error: no valid set of coefficients has been found: please supply starting values
#logit: Okay
#probit: Okay
#cauchit: Warning: glm.fit: algorithm did not converge
#cloglog: Warning: glm.fit: algorithm did not converge
# Hence, we will compare probit and logit
model.3a.logit<-glm(Growth ~ pH + Nisin + Temperature + Brix, family = binomial("logit"), data2)
model.3a.probit<-glm(Growth ~ pH + Nisin + Temperature + Brix, family = binomial("probit"), data2)

AIC(model.3a.logit)
```

```
## [1] 62.33065
```

```
AIC(model.3a.probit)
```

```
## [1] 62.01991
```

```
# Choose probit link function with Ber distribution
```

```
#b
newdata<-data.frame(pH = 4.5, Nisin = 20, Temperature = 30, Brix = 17)
pred<-predict(model.3a.probit,newdata=newdata, type="response")
pred
```

```
##         1
## 0.1011812
```

```
#c
eta<-predict(model.3a.probit, newdata=newdata, type="link", se.fit=TRUE)
link.lowerbound<-eta$fit-qnorm(0.975)*eta$se.fit
link.upperbound<-eta$fit+qnorm(0.975)*eta$se.fit
pnorm(eta$fit)
```

```
##         1
## 0.1011812
```

```
#0.1011812

mu.lowerbound<-pnorm(link.lowerbound)
mu.upperbound<-pnorm(link.upperbound)
mu.lowerbound
```

```
##          1
## 0.01412712
```

```
#0.01412712
mu.upperbound
```

```
##         1
## 0.3609346
```

```
#0.3609346
```

```
#d
options(warn=-1)
newdata<-data.frame(pH = 4.5, Nisin = 20, Temperature = 30, Brix = 17)
mu.f<-predict(model.3a.probit, newdata=newdata, type="response")
YS.pred<-100*mu.f

mu.hat<-predict(model.3a.probit, newdata=data2, type="response")

N<-dim(data2)[1]


e.b<-numeric()

for(b in 1:1000){
```

```r
yb<-numeric()
for(i in 1:N){

yb[i]<-sample(0:1,1,prob=c(1-mu.hat[i],mu.hat[i]))

}

model.b<-glm(yb[1:N]~pH + Nisin + Temperature + Brix, family = binomial("probit"), data=data2)

newdata<-data.frame(pH = 4.5, Nisin = 20, Temperature = 30, Brix = 17)

mu.fB<-predict(model.b, newdata=newdata, type="response")
YS.predB<-100*mu.fB

yf.b<-sample(0:1,100,prob=c(1-mu.f,mu.f), replace=TRUE)

e.b[b]<-sum(yf.b)-YS.predB

}

var.error<-var(e.b)
var.error
```

```
## [1] 71.36924
```

```r
z<-qnorm(c(0.9))
lower.bound<-YS.pred-z*sqrt(var.error)
upper.bound<-YS.pred+z*sqrt(var.error)
lower.bound
```

```
##          1
## -0.7084714
```

```r
upper.bound
```

```
##       1
## 20.9447
```